Coding Issues

Descriptor Area
- Infx: EXEC SQL DESCRIBE sid INTO sqlda;
 - sid is the prepared statement "SELECT * FROM tablename;"
 - the statement is never executed
 - the descriptor area is used to get at the field descriptions

- psql has not implemented a DESCRIBE statement
 - need to execute a FETCH for a record into the descriptor
  area and then access the individual fields of the record
 - does not work for empty tables
 - there is email chatter about implementing a DESCRIBE statement in the
  future

Cursors
The code generator and other embedded SQL code use a cursor for retrieving records.
Infx has EXEC SQL CLOSE and EXEC SQL FREE statements to close a cursor and free
its resources. psql has the EXEC SQL CLOSE statement which according to the doc
frees all resources related to the cursor. The psql SQL doc has no FREE statement listed.

Informix allows a cursor to be declared and opened in one routine and fetched in a
different routine. PostgreSQL does not allow this. A "cursor undefined" error is
generated by the precompiler in PostgreSQL.

Date/Time Utility Functions
- many functions in timeutil library call Infx functions such as dtcvasc for which there is
 no equivalent in psql/ecpg

- some Infx functions are not needed in psql because the "+" and "-" operators
  work for timestamp data types

- OHD will be creating new library functions to work with the generated code

Postgres Key Words
The following is a list of column and table names from the IHFS db which are postgres or
SQL standard "key words":

action
admin
date
level
location
name
notify
owner

source
state
storage
type
value

Using these key words in SQL statements may cause problems as noted below.

Informix esql statements using the key word "value" such as

EXEC SQL SELECT   value   FROM …

will generate a syntax error and must be rewritten as

EXEC SQL SELECT "value" FROM …

Note that if the above statement is written using single quotes as in

EXEC SQL SELECT 'value' FROM …

there is no syntax error. However, a runtime error of -206 is generated.

Note also that if "value" is used in a prepared statement, the double quotes (") are not necessary.

The statement

EXEC SQL SELECT county INTO :county FROM Location …

compiles and executes properly without quoting the key word "location".

The statement

EXEC SQL SELECT state INTO :state FROM Location …

compiles and executes properly without quoting the key words "state" and "location".

The statement

EXEC SQL SELECT focalpoint INTO :fp FROM Admin …

compiles and executes properly without quoting the key word "admin".

The statement

EXEC SQL SELECT level INTO :level FROM RiverStat …

compiles and executes properly without quoting the key word "level".

According to the documentation, quoting a key word makes it case-sensitive. Unquoted key words in postgres are folded into lower case. Note that folding key words into lower case is against the SQL standard. Appendix C of the postgres online documentation contains the full list of key words.